EAE

RANDOM EXERCISER

MD-11-DZKEC-A

EP-DZKEC-A-DL-A

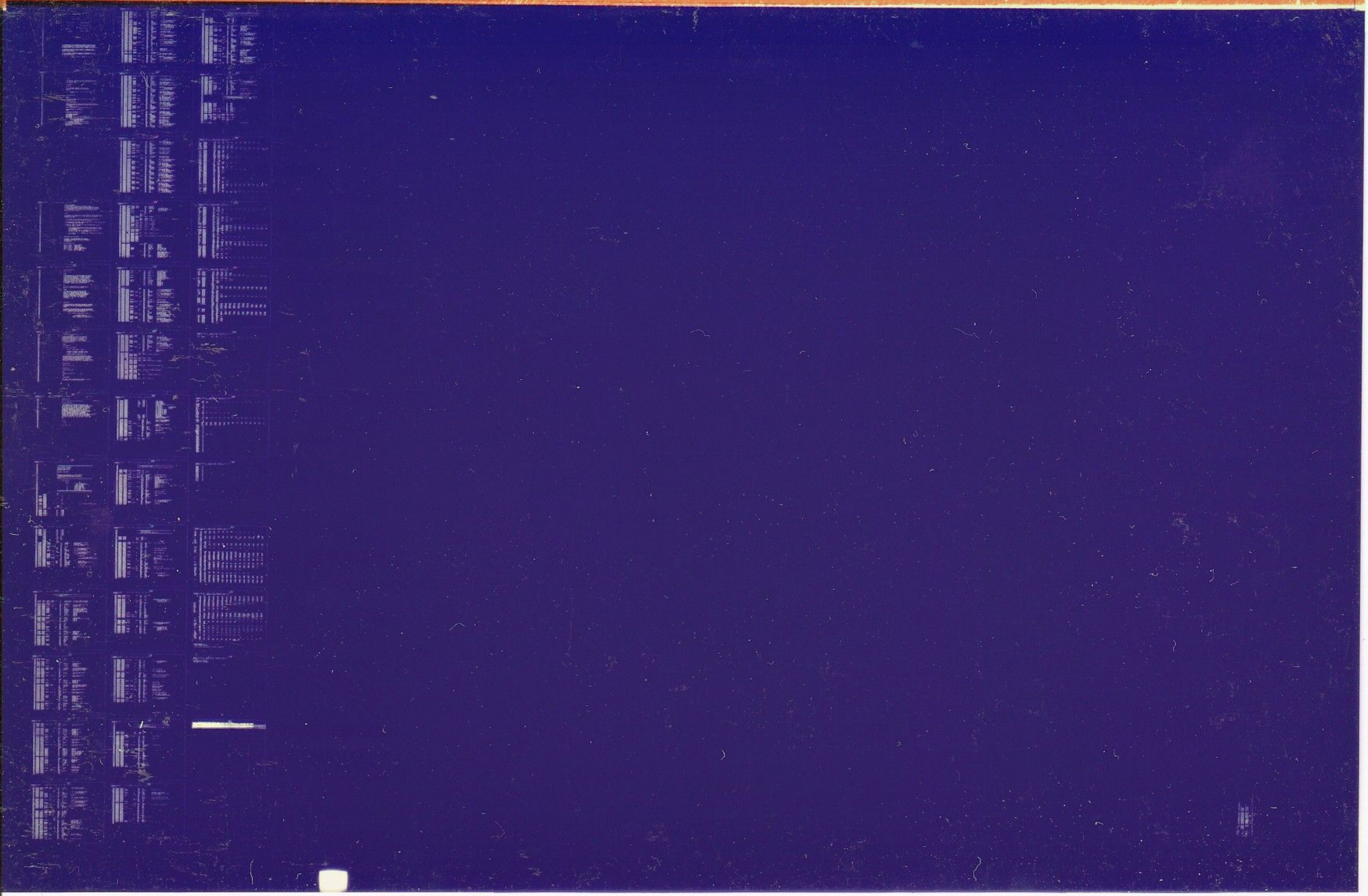
COPYRIGHT © 1976

FICHE 1 OF 1

NOV 1976

digital

MADE IN USA



```
.REM @
```

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZKEC-A
PRODUCT NAME:	EAE RANDOM EXERCIZER
DATE:	AUGUST 1976
MAINTAINER:	DIAGNOSTIC GROUP

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) BY DIGITAL EQUIPMENT CORPORATION 1970,1976

```
                                1.      ABSTRACT

                                        THIS TEST IS TO BE USED AS AN EAE EXERCIZER FOR THE PDP-11
                                        WITH THE EAE OPTION. IT TESTS THE DIVIDE AND MULTIPLY
                                        WITH RANDOM NUMBERS.

                                2.      REQUIREMENTS

                                2.1     EQUIPMENT

                                        PDP-11 STANDARD COMPUTER WITH EAE OPTION
                                        WITH OR WITHOUT HARDWARE SWITCH REGISTER
                                2.2     STORAGE

                                        2.2.1  PROGRAM STORAGE - THE ROUTINE USES MEMOR
                                               0-7020


                                3.      LOADING PROCEDURE

                                3.1     METHOD

                                        PROCEDURE FOR NORMAL ABSOLUTE TAPES SHOULD BE FOLLOWED.
                                4.      STARTING PROCEDURE

                                4.1     CONTROL SWITCH SETTING

                                        STARTING AT SA 200 ALL SWITCHES SHOULD BE SET AS INDICATED.
                                        ***IF SOFTWARE SWITCH REGISTER IS SELECTED THE FOLLOWING WILL
                                        BE PRINTED;
                                                SWR=XXXXXX NEW=
                                        REFER TO SECTION 5.1.  FOR MORE INFORMATION***

                                4.2     STARTING ADDRESS OR ADDRESSES
                                        SA=200

                                4.3     PROGRAM AND/OR OPERATOR ACTION

                                        LOAD PROGRAM INTO MEMORY.
                                        LOAD STARTING ADDRESS
                                        LOAD ADDRESS.
                                        SET SWITCHES (SEE 5) ALL DOWN FOR WORSE CASE
                                        PRESS START.
                                                NOTE: IF SOFTWARE SWITCH REGISTER IS SELECTED THEN THE
                                                FOLLOWING WILL BE PRINTED:
                                                        SWR=XXXXXX NEW=
                                                (REFER TO SECTION FOR MORE DETAILS)
                                        THE PROGRAM WILL LOOP AND
                                        BELL WILL RING ONCE PER PASS OF THE PROGRAM.
                                        A MINIMUM OF TWO PASSES SHOULD
                                        ALWAYS BE RUN.
```

```
108
```

5. OPERATING PROCEDURE
IF THE DIAGNOSTIC IS RUN ON A CPU WITHOUT A SWITCH
REGISTER THEN A SOFTWARE SWITCH REGISTER IS USED WHICH ALLOWS
THE USER THE SAME SWITCH OPTIONS AS THE HARDWARE SWITCH REGISTER.
IF THE HARDWARE SWITCH REGISTER DOES NOT EXIST OR IF ONE DOES
AND IT CONTAINS ALL ONES (177777) THEN THE SOFTWARE SWITCH
REGISTER (LOC. 176) IS USED.

CONTROL:

THIS PROGRAM ALSO SUPPORTS THE DYNAMIC LOADING OF THE SOFTWARE SWITCH
REGISTER (LOC. 176) FROM THE TTY. THIS CAN BE ACCOMPLISHED BY
DOING THE FOLLOWING:

1) TYPE CONTROL G (↑G); THIS WILL ALLOW THE TTY TO ENTER DATA INTO
   LOC. 176 AT SELECTED POINTS WITHIN THE PROGRAM.

2) THE MACHINE WILL THEN TYPE: SWR=XXXXXXNEW= (XXXXXX IS THE OCTAL CONTENTS
   OF THE SOFTWARE SWITCH REGISTER.)

3) AFTER THE ''NEW='' HAS BEEN TYPED THEN THE OPERATOR CAN DO ONE
   OF THE FOLLOWING AT THE TTY:

   A) TYPE A NUMBER TO BE LOADED INTO LOC. 176 FOLLOWED BY A (CR).
      (ONLY NUMBERS BETWEEN 0-7 WILL BE ACCEPTED AND ONLY 6 NUMBERS
      WILL BE ALLOWED)
      IF A (CR) IS THE FIRST KEY DEPRESSED THE SOFTWARE SWITCH
      REGISTER CONTENTS WILL NOT BE CHANGED.

   B) IF A CONTROL U (↑U) IS DEPRESSED THEN THE PROGRAM WILL SEND YOU
      BACK TO STEP 2.

5.1 OPERATIONAL SWITCH SETTINGS

5.1.1 AT SA 200 .. ALL SWITCHES DOWN WILL TEST ALL
FUNCTIONS OF THE EAE AND PRINT OUT ON ERRORS AND
CONTINUE IN TEST. (BELL WILL RING AT COMPLETION
OF 1000 PASS)

5.1.2 SWITCH SETTINGS ARE

SW15 = 1 OR UP ... HALT ON ERROR
SW14 = 1 OR UP ... SCOPE LOOP
SW13 = 1 OR UP ... INHIBIT PRINTOUT
SW12 = 1 OR UP ... INHIBIT TRACE TRAPPING
SW11 = 1 OR UP ... INHIBIT ITERATION LOOP
SW10 = 1 OR UP ... BELL ON ERROR
       0 OR DOWN . BELL ON 1000 PASSES

5.2. SUBROUTINE ABSTRACTS

5.2.1 BEGIN SA 200

5.2.2 SCOPE

THIS SUBROUTINE CALL IS PLACED BETWEEN EACH SUBTEST IN THE INSTRUCTION SECTION. IT RECORDS THE STARTING ADDRESS OF EACH SUB-TEST AS IT IS BEING ENTERED. IF A SCOPE LOOP IS REQUESTED, IT WILL JUMP TO THE START OF THE SUBTEST THAT THE SCOPE LOOP IS REQUESTED FOR. IF SCOPE LOOP IS NOT REQUESTED, THERE WILL BE 100 ITERATIONS ON THAT SUBTEST BEFORE THE NEXT SUBTEST IS ENTERED. SWITCH 11 ON A 1 INHIBITS ITERATION OF SUBTESTS. SUPPORTS THE CONT-G ROUTINE

5.2.3 HLT

IS A ROUTINE THAT PRINTS-OUT AN ERROR MESSAGE. SEE 6.1. SUPPORTS THE CONT-G ROUTINE

5.2.4 TRTRAP

THIS ROUTINE WILL ALLOW THE TRACE BIT TRAP TO BE SET AT THE BEGINNING OF THE PROGRAM. WHEN SET IT CAUSES A TRAP AFTER EACH INSTRUCTION. THE FIRST INSTRUCTION EXECUTED UPON TRAPPING IS AN "RTI" WHICH RETURNS TO THE INTERRUPTED SEQUENCE OF INSTRUCTION. THIS SEQUENCE IS CONTINUED TILL THE END OF THE PROGRAM LOOP IS REACHED.

5.2.5 TRAPCATCHER

THIS IS A SERIES OF INSTRUCTIONS STARTING AT LOCATION 0, DESIGNED TO DETECT, AND ISOLATE UNEXPECTED TRAPS AND INTERRUPTS TO THE TRAP AND INTERRUPT VECTOR AREA OF MEMORY.

THE PRINCIPAL OF THIS ROUTINE IS: THE VECTOR ENTERANCE CONTAINS THE ADDRESS OF THE TRAP HANDLING ROUTINE AND THE NEXT LOCATION CONTAINS THE ADDRESS TRAPPED TO DIVIDED BY 4. A MESSAGE WILL BE TYPED AS FOLLOWS:

TRAP TO LOCATION 4 FROM 3172 BY 177302

```
     4 = ADDRESS TRAPPED TO
  3172 = LOCATION THAT CAUSED THE TRAP
177302 = CONTENTS OF THAT LOCATION
```

5.3 PROGRAM AND/OR OPERATOR ACTION

5.3.1 LOADING AND STARTING AT 200 WITH ALL SWITCHES DOWN IS WORSE CASE TESTING. IF AN ERROR IS DETECTED HERE, THERE WILL BE A PRINTOUT. WHEN AN ERROR IS DETECTED AND IT IS NECESSARY TO SCOPE ON IT, PLACE SW15 UP TO HALT ON ERROR, HIT CONTINUE WITH SW14 UP TO LOOP ON ERROR, AND SW13 UP TO DELETE PRINTOUTS.

6. ERRORS

6.1 ERROR PRINTOUT

THE FORMAT IS AS FOLLOWS:

```
ERROR:   040000 000000 / 100000

SOFTWARE  AC=000000  MQ=100000  SR=22
HARDWARE  AC=000000  MQ=100000  SR=122

ERRORS OCCURED 75 TIMES OUT OF 100
```

THE LINE STARTING WITH ERROR IS THE OPERATION BEING PREFORMED WHEN THE FAILURE WAS DETECTED. THE 'SOFTWARE' LINE SHOWS THE CALCULATED RESULT AND THE 'HARDWARE' LINE IS THE RESULT BY THE HARDWARE EAE. THE LAST LINE REFLECTS THE NUMBER OF ERRORS OUT OF THE NUMBER OF TRIES. THIS LINE WILL ONLY BE TYPED IF THE ITTERATIONS ARE ON.

6.2 ERROR RECOVERY

RESTART AT 200

7. RESTRICTIONS

7.1 STARTING RESTRICTION

NONE

7.2 OPERATIONAL RESTRICTION

NONE

8. MISCELLANEOUS

THIS TEST IS ONLY AN EXERCIZER SO MAINDEC-11-DZKEB-A IS ASSUMED TO HAVE RUN ON THE EAE FIRST.

8.1 EXECUTION TIME

ABOUT 30 SECONDS WITH ALL SWITCHES DOWN

9. PROGRAM DESCRIPTION

THIS PROGRAM IS AN EXERCIZER OF THE MULTIPLY AND
DIVIDE INSTRUCTIONS. IT FIRST GENERATES 2 RANDOM NUMBERS
AND MULTIPLIES THEM TOGETHER. IT SAVES THIS ANSWER
AND DIVIDES IT BY THE MULTIPLIER AND THE MULTIPLICAN
SAVING BOTH ANSWERS AS IT GOES ALONG. THE ORIGINAL
NUMBERS ARE HARDWARE MULTIPLIED BOTH WAYS AND
COMPAIRED TO THE ORIGINAL ANSWER. THE TWO DIVISIONS
ARE PREFORMED AND COMPAIRED TO THE ORIGINAL SOFTWARE
ANSWERS. ONE MORE RANDOM NUMBER IS GENERATED
AND THIS NUMBER IS DIVIDED INTO THE ORIGINAL TWO
NUMBERS AND THE ANSWER CHECKED AGAINST THE HARDWARE
ANSWERS. EACH HARDWARE MULTIPLY AND DIVIDE IS DONE
100 TIMES. IF SW11 IS UP, THEY WILL ONLY BY DONE ONCE.
SW14 WILL CAUSE THE TEST THAT YOU ARE IN TO BE LOOPED
UPON.

10. LISTING

FOLLOWING

11. FLOW CHART(S)

NA

@

```
                        ;;*****************************************************************
                        ;;*****************************************************************
                        .TITLE  MAINDEC-11-DZKEC-A
                        ;*COPYRIGHT (C) 1970,1976
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*PROGRAM BY BOB BRAIN
                        ;*
                        ;*
                        ;*MODIFIED BY ALAN BOSTICK JULY 1976 TO SUPPORT
                        ;*SOFTWARE SWITCH REGISTER, ALSO ALLOWS THE
                        ;*DYNAMIC LOADING OOF SOFTWARE SWITCH REGISTER
                        ;*FROM TTY
                        ;;*****************************************************************

                        ;       SWITCH SETTINGS

                        ;               SWITCH          USE
                        ;               ------          --------------------
                        ;                 10            0 - BELL ON PASS COMPLETE
                        ;                               1 - BELL ON ERROR
                        ;                 11            INHIBIT ITERATIONS
                        ;                 12            INHIBIT TRACE TRAP
                        ;                 13            INHIBIT ERROR TYPEOUTS
                        ;                 14            LOOP ON TEST
                        ;                 15            HALT ON ERROR
                        ;;*****************************************************************
                        ;;*****************************************************************
                        ;;*****************************************************************

                        .ABS
                        .=0
                        .=14
000014  004474                  .TRT
000016  000000                  0
                        .=30
000030  002302                  .EMT
000032  000000                  0
000034  002646                  .TRP
000036  000000                  0
                        .=64
000064  004766                  TYPER                   ;SET TTY PRINTER INTERRUPT VECTOR
000066  000000                  0
                        .=176
000176  000000          SWREG:  0
                        .=200
000200  000167  000574          JMP     BEGIN

000204  177300          DIV:    177300
000206  177302          AC:     177302
000210  177304          MQ:     177304
000212  177306          MUL:    177306
000214  177310          SC:     177310
000216  177311          SR:     177311
000220  177312          NOR:    177312
```

```
                                        LSH:    177314
                                        ASH:    177316
                                        SWR:    177570
                                        TKS:    177560
                                        TKB:    177562
                                        TPS:    177564
                                        TPB:    177566
                                        TKV=    60
                                        TPV=    64
                                        SCOPE=  EMT
                                        HLT=    TRAP
                                        .AC=    %0
                                        .MQ=    %1
                                        .=1000
001000  012706  001000          BEGIN:  MOV     #BEGIN,%6       ;SET PUSH DOWN LIST POINTER
001004  005067  003754                  CLR     ERRORS          ;ERROR COUNT
001010  005067  176762                  CLR     177776          ;CLEAR PSW, ENABLE INTERUPT
001014  005003                          CLR     %3              ;ZERO PRINT REGISTER
001016  012767  000001  003706          MOV     #1,CNT
001024  005067  003700                  CLR     CINT
001030  005703                  64$:    TST     %3              ;WAIT FOR OLD PRINTING SEQUENCE
001032  001376                          BNE     64$             ;       TO BE COMPLETED
001034  012703  004623                  MOV     #HEAD,%3        ;INITIALIZE PRINT REGISTER
001040  052777  000100  177166          BIS     #100,@TPS
001046  005703                  65$:    TST     %3              ;WAIT FOR OLD PRINTING SEQUENCE
001050  001376                          BNE     65$             ;       TO BE COMPLETED
001052  012703  004655                  MOV     #MAIN,%3        ;INITIALIZE PRINT REGISTER
001056  052777  000100  177150          BIS     #100,@TPS
001064  013746  000006          SUSWR:  MOV     @#6,-(SP)               ;SAVE VECTORS
001070  013746  000004                  MOV     @#4,-(SP)
001074  012737  001114  000004          MOV     #64$,@#4                ;SET UP FOR TIMEOUT
001102  022777  177777  177116          CMP     #-1,@SWR        ;REFERENCE HARDWARE SWITCH REGISTER
001110  001402                          BEQ     65$
001112  000404                          BR      66$
001114  022626                  64$:    CMP     (SP)+,(SP)+     ;ADJUST STACK
001116  012767  000176  177102  65$:    MOV     #SWREG,SWR      ;POINT TO SOFTWARE SWITCH REG
001124  012637  000004          66$:    MOV     (SP)+,@#4       ;RESTORE VECTORS
001130  012637  000006                  MOV     (SP)+,@#6
001134  022767  000176  177064          CMP     #SWREG,SWR      ;IS SWREG USED
001142  001002                          BNE     67$
001144  004767  005250                  JSR     PC,CNTLU        ;ALLOW SWREG TO BE LOADED
```

```
                                ;;*****************************************************************
                                ;       BEGIN TESTING
                                ;;*****************************************************************


001150  012767  123456  003546  67$:    MOV     #123456,LONUM   ;INITALIZE LONUM FOR RANDOM
001156  012767  176543  003542          MOV     #176543,HINUM   ;INITALIZE HINUM FOR RANDOM

001164  004767  005150          GO:     JSR     %7,CKSWR
001170  032777  010000  177030          BIT     #10000,@SWR     ;BIT 12=1 INHIBIT TRACE TRAP
001176  001004                          BNE     1$              ;BRANCH IF NO
001200  052767  000020  176570          BIS     #20,177776      ;SET PSW T BIT IF YES
001206  000402                          BR      2$
001210  005067  176562          1$:     CLR     177776          ;CLEAR PSW, NO TRAP TRACE
001214  012704  004734          2$:     MOV     #AREA,%4        ;SET POINTER TO ANSWER AREA
001220  004767  002626                  JSR     %7,RANDOM       ;GET RANDOM NUMBERS
001224  016701  003474                  MOV     LONUM,.MQ       ;LOAD MULTIPLIER (SOFTWARE)
001230  016702  003472                  MOV     HINUM,%2        ;LOAD MULTIPLICAN (SOFTWARE)
001234  004767  004570                  JSR     %7,MUL1         ;MULTIPLY
001240  010124                          MOV     .MQ,(4)+        ;SAVE MQ
001242  010024                          MOV     .AC,(4)+        ;SAVE AC
001244  016724  004002                  MOV     SR1,(4)+        ;SAVE SR

001250  016702  003450                  MOV     LONUM,%2        ;LOAD DIVISOR
001254  004767  003776                  JSR     %7,DIVI         ;DIVIDE BY MULTIPLIER (SOFTWARE)
001260  010024                          MOV     .AC,(4)+        ;SAVE AC
001262  010124                          MOV     .MQ,(4)+        ;SAVE MQ
001264  016724  003762                  MOV     SR1,(4)+        ;SAVE SR

001270  005767  003430                  TST     LONUM
001274  001411                          BEQ     OK1
001276  005700                          TST     .AC
001300  001402                          BEQ     3$
001302  004767  002670                  JSR     %7,BAD
001306  020167  003414          3$:     CMP     .MQ,HINUM
001312  001402                          BEQ     OK1
001314  004767  002656                  JSR     %7,BAD

001320  012704  004734          OK1:    MOV     #AREA,%4        ;RESET POINTER
001324  012401                          MOV     (4)+,.MQ        ;RESTORE MQ
001326  012400                          MOV     (4)+,.AC        ;RESTORE AC
001330  062704  000010                  ADD     #10,%4          ;INCREMENT POINTER

001334  016702  003366                  MOV     HINUM,%2        ;LOAD DIVISOR
001340  004767  003712                  JSR     %7,DIVI         ;DIVIDE BY MULTIPLICAN (SOFTWARE)
001344  010024                          MOV     .AC,(4)+        ;SAVE AC
001346  010124                          MOV     .MQ,(4)+        ;SAVE MQ
001350  016714  003676                  MOV     SR1,(4)         ;SAVE SR
001354  005767  003346                  TST     HINUM
001360  001411                          BEQ     ITER
001362  005700                          TST     .AC
001364  001402                          BEQ     1$
001366  004767  002604                  JSR     %7,BAD
001372  020167  003326          1$:     CMP     .MQ,LONUM
```

```
  453  001376  001402                          BEQ     ITER
  454  001400  004767  002572                  JSR     %7,BAD
  455
  456  001404  005067  003320          ITER:   CLR     CINT            ;SET UP COUNT
  457  001410  104000                          SCOPE
  458  001412  012704  004734                  MOV     #AREA,%4        ;RESTORE POINTER
  459  001416  012401                          MOV     (4)+,.MQ        ;RESTORE MQ
  460  001420  012400                          MOV     (4)+,.AC        ;RESTORE AC
  461  001422  011467  003624                  MOV     (4),SR1         ;RESTORE SR1
  462
  463  001426  016777  003272  176554          MOV     LONUM,@MQ       ;LOAD MULTIPLIER (HARDWARE)
  464  001434  016777  003266  176550          MOV     HINUM,@MUL      ;LOAD MULTIPLICAN AND MULTIPLY
  465  001442  020077  176540                  CMP     .AC,@AC         ;CHECK HIGH ORDER PRODUCT
  466  001446  001401                          BEQ     1$              ;SKIP IF OK
  467  001450  104400                          HLT
  468  001452  020177  176532          1$:     CMP     .MQ,@MQ         ;CHECK LOW ORDER PRODUCT
  469  001456  001401                          BEQ     2$              ;SKIP IF OK
  470  001460  104400                          HLT
  471  001462  126777  003564  176526  2$:     CMPB    SR1,@SR         ;CHECK STATUS REGISTER
  472  001470  001401                          BEQ     3$              ;SKIP IF GOOD
  473  001472  104400                          HLT
  474
  475  001474  104000                  3$:     SCOPE
  476  001476  012704  004734                  MOV     #AREA,%4        ;RESTORE POINTER
  477  001502  012401                          MOV     (4)+,.MQ        ;RESTORE MQ
  478  001504  012400                          MOV     (4)+,.AC        ;RESTORE AC
  479  001506  011467  003540                  MOV     (4),SR1         ;RESTORE SR1
  480
  481  001512  016777  003210  176470          MOV     HINUM,@MQ       ;LOAD MULTIPLIER (HARDWARE)
  482  001520  016777  003200  176464          MOV     LONUM,@MUL      ;LOAD MULTIPLICN AND MULTIPLY
  483  001526  020077  176454                  CMP     .AC,@AC         ;CHECK HIGH ORDER PRODUCT
  484  001532  001401                          BEQ     4$              ;SKIP IF OK
  485  001534  104402                          HLT+2
  486  001536  020177  176446          4$:     CMP     .MQ,@MQ         ;CHECK LOW ORDER PRODUCT
  487  001542  001401                          BEQ     5$              ;SKIP IF OK
  488  001544  104402                          HLT+2
  489  001546  126777  003500  176442  5$:     CMPB    SR1,@SR         ;CHECK STATUS REGISTER
  490  001554  001401                          BEQ     6$              ;SKIP IF GOOD
  491  001556  104402                          HLT+2
  492  001560  104000                  6$:     SCOPE
  493  001562  012704  004734                  MOV     #AREA,%4        ;RESTORE POINTER
  494  001566  012477  176416                  MOV     (4)+,@MQ        ;RESTORE MQ
  495  001572  012477  176410                  MOV     (4)+,@AC        ;RESTORE AC
  496  001576  005724                          TST     (4)+            ;INTREMENT POINTER
  497  001600  012400                          MOV     (4)+,.AC        ;RESTORE AC
  498  001602  012401                          MOV     (4)+,.MQ        ;RESTORE MQ
  499  001604  011467  003442                  MOV     (4),SR1         ;RESTORE SR1
  500
  501  001610  016777  003110  176366          MOV     LONUM,@DIV      ;DIVIDE BY MULTIPLIER (HARDWARE)
  502  001616  004767  000414                  JSR     %7,FLOW
  503  001622  000410                          BR      .SR1
  504  001624  020077  176356                  CMP     .AC,@AC         ;CHECK REMAINDER
  505  001630  001401                          BEQ     7$              ;SKIP IF OK
  506  001632  104401                          HLT+1
  507  001634  020177  176350          7$:     CMP     .MQ,@MQ         ;CHECK QUOTENT
  508  001640  001401                          BEQ     .SR1            ;SKIP IF OK
```

```
                                                HLT+1
001642  104401
001644  126777  003402  176344  .SR1:   CMPB    SR1,@SR         ;CHECK STATUS REGISTER
001652  001401                          BEQ     8$              ;SKIP IF OK
001654  104401                          HLT+1

001656  104000                  8$:     SCOPE
001660  012704  004734                  MOV     #AREA,%4        ;RESTORE POINTER
001664  012477  176320                  MOV     (4)+,@MQ        ;RESTORE MQ
001670  012477  176312                  MOV     (4)+,@AC        ;RESTORE AC
001674  062704  000010                  ADD     #10,%4          ;INCREMENT POINTER
001700  012400                          MOV     (4)+,.AC        ;RESTORE AC
001702  012401                          MOV     (4)+,.MQ        ;RESTORE MQ
001704  011467  003342                  MOV     (4),SR1         ;RESTORE SR1

001710  016777  003012  176266          MOV     HINUM,@DIV      ;DIVIDE BY MULTIPLICAN (HARDWARE)
001716  004767  000314                  JSR     %7,FLOW
001722  000410                          BR      .SR2
001724  020077  176256                  CMP     .AC,@AC         ;CHECK REMAINDER
001730  001401                          BEQ     9$              ;SKIP IF OK
001732  104403                          HLT+3
001734  020177  176250          9$:     CMP     .MQ,@MQ         ;CHECK QUOTENT
001740  001401                          BEQ     .SR2            ;SKIP IF OK
001742  104403                          HLT+3
001744  126777  003302  176244  .SR2:   CMPB    SR1,@SR         ;CHECK STATUS REGISTER
001752  001401                          BEQ     1$              ;SKIP IF OK
001754  104403                          HLT+3
001756  104000                  1$:     SCOPE
001760  012704  004734                  MOV     #AREA,%4        ;RESET POINTER
001764  016724  002734                  MOV     LONUM,(4)+      ;SAVE MQ
001770  016724  002732                  MOV     HINUM,(4)+      ;SAVE AC
001774  016701  002724                  MOV     LONUM,.MQ       ;LOAD MQ (SOFTWARE)
002000  016700  002722                  MOV     HINUM,.AC       ;LOAD AC (SOFTWARE)
002004  004767  002042                  JSR     %7,RANDOM       ;GET 1 RANDOM NUMBER
002010  016714  002710                  MOV     LONUM,(4)       ;SAVE THE DIVISOR
002014  016702  002704                  MOV     LONUM,%2        ;LOAD THA DIVISOR
002020  004767  003232                  JSR     %7,DIVI         ;DIVIDE

002024  005067  002700                  CLR     CINT            ;SET COUNTER
002030  104000                          SCOPE
002032  012704  004734                  MOV     #AREA,%4        ;RESTORE POINTER
002036  012477  176146                  MOV     (4)+,@MQ        ;RESTORE MQ
002042  012477  176140                  MOV     (4)+,@AC        ;RESTORE AC
002046  011477  176132                  MOV     (4),@DIV        ;DIVIDE
002052  004767  000160                  JSR     %7,FLOW
002056  000410                          BR      .SR3
002060  020077  176122                  CMP     .AC,@AC         ;CHECK AC
002064  001401                          BEQ     2$
002066  104407                          HLT+7                   ;REMAINDER WRONG
002070  020177  176114          2$:     CMP     .MQ,@MQ         ;CHECK QUOTIENT
002074  001401                          BEQ     .SR3
002076  104407                          HLT+7                   ;QUOTIENT WRONG
002100  126777  003146  176110  .SR3:   CMPB    SR1,@SR         ;CHECK STATUS
002106  001401                          BEQ     1$
002110  104407                          HLT+7                   ;STATUS WRONG
002112  104000                  1$:     SCOPE
```

```
   565  002114  062767  000001  002610          ADD     #1,CNT          ;COUNT NUMBER OF PASSES
   566  002122  032767  000777  002602          BIT     #777,CNT        ;CHECK FOR MULTIPLES OF 1000
   567  002130  001411                          BEQ     3$
   568  002132                          2$:
   569  002132  005703                  64$:    TST     %3              ;WAIT FOR OLD PRINTING SEQUENCE
   570  002134  001376                          BNE     64$             ;       TO BE COMPLETED
   571  002136  012703  004704                  MOV     #EAEOK,%3       ;INITIALIZE PRINT REGISTER
   572  002142  052777  000100  176064          BIS     #100,@TPS
   573  002150  000167  177010                  JMP     GO              ;RERUN
   574  002154  004767  004160          3$:     JSR     %7,CKSWR
   575  002160  032777  002000  176040          BIT     #2000,@SWR      ;BIT 10=1 BELL ON ERROR
   576                                                                  ;BIT 10=0 BELL ON PASS COMPLETE
   577  002166  001361                          BNE     2$
   578  002170  112767  000007  002564          MOVB    #7,.TYPE        ;TYPE A 7
   579  002176  005703                  65$:    TST     %3              ;WAIT FOR OLD PRINTING SEQUENCE
   580  002200  001376                          BNE     65$             ;       TO BE COMPLETED
   581  002202  012703  004762                  MOV     #.TYPE,%3       ;INITIALIZE PRINT REGISTER
   582  002206  052777  000100  176020          BIS     #100,@TPS
   583  002214  005703                  66$:    TST     %3              ;WAIT FOR OLD PRINTING SEQUENCE
   584  002216  001376                          BNE     66$             ;       TO BE COMPLETED
   585  002220  012703  004704                  MOV     #EAEOK,%3       ;INITIALIZE PRINT REGISTER
   586  002224  052777  000100  176002          BIS     #100,@TPS
   587  002232  000167  176726                  JMP     GO              ;RERUN
   588
   589
   590  002236  117767  175754  000034  FLOW:   MOVB    @SR,TEMP
   591  002244  142767  000077  000026          BICB    #77,TEMP
   592  002252  122767  000200  000020          CMPB    #200,TEMP
   593  002260  001406                          BEQ     EQZ
   594  002262  122767  000100  000010          CMPB    #100,TEMP
   595  002270  001402                          BEQ     EQZ
   596  002272  062716  000002                  ADD     #2,(6)
   597  002276  000207                  EQZ:    RTS     %7
   598
   599  002300  000000                  TEMP:   0
   600  002302  005767  002422          .EMT:   TST     CINT            ;CHECK FOR FIRST ONE
   601  002306  001013                          BNE     NXT1            ;SKIP IF FIRST
   602  002310  005267  002414                  INC     CINT            ;DECREASE THE COUNTER
   603  002314  004767  004020                  JSR     %7,CKSWR        ;CHECK FOR CONT-G
   604  002320  032777  040000  175700          BIT     #40000,@SWR     ;BIT 14=1 LOOP ON TEST
   605  002326  001142                          BNE     OVR
   606  002330  011667  002424                  MOV     (6),LAD         ;SAVE LOOP ADDRESS
   607  002334  000002                          RTI                     ;RETURN
   608  002336  004767  003776          NXT1:   JSR     %7,CKSWR        ;CHECK FOR CONT-G
   609  002342  032777  004000  175656          BIT     #4000,@SWR      ;BIT 11=1 INHIBIT ITERATION LOOP
   610  002350  001405                          BEQ     KIT
   611  002352  005067  002400                  CLR     ERCNT
   612  002356  005067  002346          CLCI:   CLR     CINT
   613  002362  000747                          BR      .EMT
   614  002364  005267  002340          KIT:    INC     CINT
   615  002370  004767  003744                  JSR     %7,CKSWR        ;CHECK FOR CONT-G
   616  002374  032777  040000  175624          BIT     #40000,@SWR     ;BIT 14=1 LOOP ON TEST
   617  002402  001114                          BNE     OVR
   618  002404  005767  002320                  TST     CINT
   619  002410  100404                          BMI     TST1
   620  002412  022767  000100  002310          CMP     #100,CINT       ;CHECK FOR LAST ONE
```

```
                                                TST1:   BPL     OVR             ;LAST ONE?
                                                        TST     ERCNT           ;CHECK FOR ANY ERRORS
                                                        BEQ     CLCI            ;NO, GO TO NEXT TEST
                                                        JSR     %7,CKSWR        ;CHECK FOR GO/NOGO
                                                        BIT     #20000,@SWR     ;BIT 13=1 INHIBIT PRINTOUT
                                                        BEQ     1$
                                                        CLR     ERCNT
                                                        BR      CLCI
                                                1$:
                                                64$:    TST     %3              ;WAIT FOR OLD PRINTING SEQUENCE
                                                        BNE     64$             ;       TO BE COMPLETED
                                                        MOV     #ER6,%3         ;INITIALIZE PRINT REGISTER
                                                        BIS     #100,@TPS
                                                        MOV     ERCNT,PRINT1    ;TYPE ERCNT IN OCTAL
                                                        JSR     %7,PRINTS       ;AND SUPRESS LEADING ZERO'S
                                                65$:    TST     %3              ;WAIT FOR OLD PRINTING SEQUENCE
                                                        BNE     65$             ;       TO BE COMPLETED
                                                        MOV     #ER7,%3         ;INITIALIZE PRINT REGISTER
                                                        BIS     #100,@TPS
                                                        CLR     ERCNT           ;CLEAR ERROR COUNTER
                                                        DEC     CINT
                                                        MOV     CINT,PRINT1     ;TYPE CINT IN OCTAL
                                                        JSR     %7,PRINTS       ;AND SUPRESS LEADING ZERO'S
                                                        CLR     CINT
                                                        MOVB    #'<,.TYPE       ;TYPE A '<
                                                66$:    TST     %3              ;WAIT FOR OLD PRINTING SEQUENCE
                                                        BNE     66$             ;       TO BE COMPLETED
                                                        MOV     #.TYPE,%3       ;INITIALIZE PRINT REGISTER
                                                        BIS     #100,@TPS
                                                DOWN:   JSR     %7,CKSWR
                                                        TST     @SWR            ;CHECK FOR HALT ON ERROR
                                                        BPL     RT
                                                64$:    TST     %3              ;WAIT FOR OLD PRINTING SEQUENCE
                                                        BNE     64$             ;       TO BE COMPLETED
                                                        MOV     #CON,%3         ;INITIALIZE PRINT REGISTER
                                                        BIS     #100,@TPS
                                                65$:    TST     %3              ;WAIT FOR THE PRINTER TO FINISH
                                                        BNE     65$
                                                        HALT
                                                RT:     JMP     .EMT
                                                OVR:    TST     (6)+            ;RESET POINTER
                                                        MOV     (6)+,177776     ;RESTORE PS
                                                        JMP     @LAD            ;JMP TO BEGINNING OF TEST

                                                .TRP:   JSR     %7,CKSWR
                                                        BIT     #2000,@SWR      ;BIT 10=1 BELL ON ERROR
                                                                                ;BIT 10=0 BELL ON PASS COMPLETE
                                                        BEQ     .ET
                                                        MOVB    #7,.TYPE        ;TYPE A 7
                                                64$:    TST     %3              ;WAIT FOR OLD PRINTING SEQUENCE
                                                        BNE     64$             ;       TO BE COMPLETED
                                                        MOV     #.TYPE,%3       ;INITIALIZE PRINT REGISTER
                                                        BIS     #100,@TPS
                                                65$:    TST     %3              ;WAIT FOR THE PRINTER TO FINISH
                                                        BNE     65$
                                                .ET:    JSR     %7,CKSWR
```

```
  677  002716  032777  040000  175302          BIT     #40000,@SWR     ;BIT 14=1 LOOP ON TEST
  678  002724  001402                          BEQ     1$
  679  002726  000167  000662                  JMP     NOHEAD
  680  002732  005767  002020          1$:     TST     ERCNT           ;CHECK FOR FIRST ERROR
  681  002736  001402                          BEQ     3$              ;NO, GO OVER HEADING PRINTER
  682  002740  000167  000650          2$:     JMP     NOHEAD
  683  002744  011667  000660          3$:     MOV     (6),HLTAD       ;SAVE ADDRESS OF HLT
  684  002750  005267  002010                  INC     ERRORS          ;COUNT THE NUMBER OF ERRORS
  685  002754  004767  003360                  JSR     %7,CKSWR        ;CHECK FOR CONT-G
  686  002760  032777  020000  175240          BIT     #20000,@SWR     ;BIT 13=1 INHIBIT ERROR TYPEOUTS
  687  002766  001364                          BNE     2$
  688  002770  112767  000137  001764          MOVB    #'←,.TYPE       ;TYPE A '←
  689  002776  005703                  64$:    TST     %3              ;WAIT FOR OLD PRINTING SEQUENCE
  690  003000  001376                          BNE     64$             ;       TO BE COMPLETED
  691  003002  012703  004762                  MOV     #.TYPE,%3       ;INITIALIZE PRINT REGISTER
  692  003006  052777  000100  175220          BIS     #100,@TPS
  693  003014  016767  001712  002042          MOV     CNT,PRINT1      ;TYPE CNT IN OCTAL
  694  003022  004767  002062                  JSR     %7,PRINTS       ;AND SUPRESS LEADING ZERO'S
  695  003026  005703                  65$:    TST     %3              ;WAIT FOR OLD PRINTING SEQUENCE
  696  003030  001376                          BNE     65$             ;       TO BE COMPLETED
  697  003032  012703  003634                  MOV     #ER1,%3         ;INITIALIZE PRINT REGISTER
  698  003036  052777  000100  175170          BIS     #100,@TPS
  699  003044  011646                          MOV     (6),-(6)        ;PUT ADDRESS OF INSTRUCTION ON STACK
  700  003046  162716  000002                  SUB     #2,(6)          ;GET THE INSTRUCTION
  701  003052  032776  000001  000000          BIT     #1,@(6)         ;TEST FOR MULTIPLY OR DIVIDE
  702  003060  001050                          BNE     DIVP            ;GOTO DIVIDE PRINT IF ON A 1
  703  003062  032776  000002  000000          BIT     #2,@(6)         ;TEST FOR WHICH MULTIPLY
  704  003070  001422                          BEQ     SECO
  705  003072  016767  001626  001764          MOV     LONUM,PRINT1    ;TYPE LONUM IN OCTAL
  706  003100  004767  001774                  JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
  707  003104  005703                  66$:    TST     %3              ;WAIT FOR OLD PRINTING SEQUENCE
  708  003106  001376                          BNE     66$             ;       TO BE COMPLETED
  709  003110  012703  003650                  MOV     #ER2,%3         ;INITIALIZE PRINT REGISTER
  710  003114  052777  000100  175112          BIS     #100,@TPS
  711  003122  016767  001600  001734          MOV     HINUM,PRINT1    ;TYPE HINUM IN OCTAL
  712  003130  004767  001744                  JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
  713  003134  000506                          BR      PSOFT
  714  003136                           SECO:
  715  003136  016767  001564  001720          MOV     HINUM,PRINT1    ;TYPE HINUM IN OCTAL
  716  003144  004767  001730                  JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
  717  003150  005703                  64$:    TST     %3              ;WAIT FOR OLD PRINTING SEQUENCE
  718  003152  001376                          BNE     64$             ;       TO BE COMPLETED
  719  003154  012703  003650                  MOV     #ER2,%3         ;INITIALIZE PRINT REGISTER
  720  003160  052777  000100  175046          BIS     #100,@TPS
  721  003166  016767  001532  001670          MOV     LONUM,PRINT1    ;TYPE LONUM IN OCTAL
  722  003174  004767  001700                  JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
  723  003200  000464                          BR      PSOFT
  724  003202  012704  004736          DIVP:   MOV     #AREA+2,%4      ;SET POINTER AT AC
  725  003206  011467  001652                  MOV     (4),PRINT1      ;TYPE (4) IN OCTAL
  726  003212  004767  001662                  JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
  727  003216  112767  000040  001536          MOVB    #40,.TYPE       ;TYPE A 40
  728  003224  005703                  64$:    TST     %3              ;WAIT FOR OLD PRINTING SEQUENCE
  729  003226  001376                          BNE     64$             ;       TO BE COMPLETED
  730  003230  012703  004762                  MOV     #.TYPE,%3       ;INITIALIZE PRINT REGISTER
  731  003234  052777  000100  174772          BIS     #100,@TPS
  732  003242  014467  001616                  MOV     -(4),PRINT1     ;TYPE -(4) IN OCTAL
```

```
  733  003246  004767  001626                 JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
  734  003252  005703                 65$:    TST     %3              ;WAIT FOR OLD PRINTING SEQUENCE
  735  003254  001376                         BNE     65$             ;       TO BE COMPLETED
  736  003256  012703  003654                 MOV     #ER2A,%3        ;INITIALIZE PRINT REGISTER
  737  003262  052777  000100  174744         BIS     #100,@TPS
  738  003270  032776  000004  000000         BIT     #4,@(6)
  739  003276  001406                         BEQ     DIVZ
  740  003300  016767  001434  001556         MOV     AREA+4,PRINT1   ;TYPE AREA+4 IN OCTAL
  741  003306  004767  001566                 JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
  742  003312  000417                         BR      PSOFT
  743  003314  032776  000002  000000 DIVZ:   BIT     #2,@(6)         ;TEST FOR WHICH DIVIDE
  744  003322  001006                         BNE     SECI
  745  003324  016767  001374  001532         MOV     LONUM,PRINT1    ;TYPE LONUM IN OCTAL
  746  003332  004767  001542                 JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
  747  003336  000405                         BR      PSOFT
  748  003340                          SECI:
  749  003340  016767  001362  001516         MOV     HINUM,PRINT1    ;TYPE HINUM IN OCTAL
  750  003346  004767  001526                 JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
  751  003352                          PSOFT:
  752  003352  005703                 64$:    TST     %3              ;WAIT FOR OLD PRINTING SEQUENCE
  753  003354  001376                         BNE     64$             ;       TO BE COMPLETED
  754  003356  012703  003660                 MOV     #ER3,%3         ;INITIALIZE PRINT REGISTER
  755  003362  052777  000100  174644         BIS     #100,@TPS
  756  003370  010067  001470                 MOV     .AC,PRINT1      ;TYPE .AC IN OCTAL
  757  003374  004767  001500                 JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
  758  003400  005703                 65$:    TST     %3              ;WAIT FOR OLD PRINTING SEQUENCE
  759  003402  001376                         BNE     65$             ;       TO BE COMPLETED
  760  003404  012703  003714                 MOV     #ER4,%3         ;INITIALIZE PRINT REGISTER
  761  003410  052777  000100  174616         BIS     #100,@TPS
  762  003416  010167  001442                 MOV     .MQ,PRINT1      ;TYPE .MQ IN OCTAL
  763  003422  004767  001452                 JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
  764  003426  005703                 66$:    TST     %3              ;WAIT FOR OLD PRINTING SEQUENCE
  765  003430  001376                         BNE     66$             ;       TO BE COMPLETED
  766  003432  012703  003724                 MOV     #ER5,%3         ;INITIALIZE PRINT REGISTER
  767  003436  052777  000100  174570         BIS     #100,@TPS
  768  003444  016767  001602  001412         MOV     SR1,PRINT1      ;TYPE SR1 IN OCTAL
  769  003452  004767  001432                 JSR     %7,PRINTS       ;AND SUPRESS LEADING ZERO'S
  770  003456  005703                 67$:    TST     %3              ;WAIT FOR OLD PRINTING SEQUENCE
  771  003460  001376                         BNE     67$             ;       TO BE COMPLETED
  772  003462  012703  003702                 MOV     #ER3B,%3        ;INITIALIZE PRINT REGISTER
  773  003466  052777  000100  174540         BIS     #100,@TPS
  774  003474  005703                 68$:    TST     %3              ;WAIT FOR OLD PRINTING SEQUENCE
  775  003476  001376                         BNE     68$             ;       TO BE COMPLETED
  776  003500  012703  003672                 MOV     #ER3A,%3        ;INITIALIZE PRINT REGISTER
  777  003504  052777  000100  174522         BIS     #100,@TPS
  778  003512  017767  174470  001344         MOV     @AC,PRINT1      ;TYPE @AC IN OCTAL
  779  003520  004767  001354                 JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
  780  003524  005703                 69$:    TST     %3              ;WAIT FOR OLD PRINTING SEQUENCE
  781  003526  001376                         BNE     69$             ;       TO BE COMPLETED
  782  003530  012703  003714                 MOV     #ER4,%3         ;INITIALIZE PRINT REGISTER
  783  003534  052777  000100  174472         BIS     #100,@TPS
  784  003542  017767  174442  001314         MOV     @MQ,PRINT1      ;TYPE @MQ IN OCTAL
  785  003550  004767  001324                 JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
  786  003554  005703                 70$:    TST     %3              ;WAIT FOR OLD PRINTING SEQUENCE
  787  003556  001376                         BNE     70$             ;       TO BE COMPLETED
  788  003560  012703  003724                 MOV     #ER5,%3         ;INITIALIZE PRINT REGISTER
```

```
                                                BIS     #100,@TPS
                                                MOVB    @SR,SER
                                                MOV     SER,PRINT1      ;TYPE SER IN OCTAL
                                                JSR     %7,PRINTS       ;AND SUPRESS LEADING ZERO'S
                                                TST     (6)+            ;RESTORE POINTER
                                        NOHEAD: CMP     (6),HLTAD       ;CHECK FOR SAME HLT
                                                BNE     1$
                                                INC     ERCNT
                                        1$:     RTI
                                        HLTAD:  0
                                        SER:    0
                                        ER1:    .ASCII  "←←ERROR:   &"
                                        ER2:    .ASCII  " X &"
                                        ER2A:   .ASCII  " / &"
                                        ER3:    .ASCII  "←←SOFTWARE"
                                        ER3A:   .ASCII  "    AC=&"
                                        ER3B:   .ASCII  "←HARDWARE&"
                                        ER4:    .ASCII  "    MQ=&"
                                        ER5:    .ASCII  "    SR=&"
                                        ER6:    .ASCII  "←←ERRORS OCCURRED &"
                                        ER7:    .ASCII  " TIMES OUT OF &"
                                        ERNG:   .ASCII  "←SOFTWARE DIVIDE DOESN'T MATCH REAL ANSWER←&"
                                                .EVEN
                                        RANDOM: MOV     .AC,-(6)        ;SAVE R0
                                                MOV     .MQ,-(6)        ;SAVE R1
                                                MOV     %2,-(6)         ;SAVE R2
                                                MOV     %4,-(6)         ;SAVE R4
                                                MOV     LONUM,.AC       ;SET R0 WITH LOW
                                                MOV     HINUM,.MQ       ;SET R1 WITH HIGH
                                                MOV     #-7,%4          ;SET SHIFT COUNT
                                                CLR     %2
                                        SHIFT:  ASL     .AC             ;SHIFT R0 LEFT AND
                                                ROL     .MQ             ;ROTATE CARRY INTO R1 AND
                                                ROL     %2              ;ROTATE CARRY INTO R2
                                                INC     %4              ;CHECK FOR DONE
                                                BNE     SHIFT           ;CONTINUE SHIFT LOOP
                                                ADD     LONUM,%2        ;ADD NUMBER TO MAKE X 129
                                                ADC     .MQ             ;PROPOGATE CARRY
```

```
                                        ADD     HINUM,.MQ       ;ADD NUMBER TO MAKE X 129
                                        ADC     %2              ;PROPOGATE CARRY
                                        ADD     #1057,.AC       ;ADD LOW CONSTANT
                                        ADC     .MQ             ;PROPOGATE CARRY
                                        ADC     %2              ;PROPOGATE CARRY
                                        ADD     #47401,.MQ      ;ADD HIGH CONSTANT
                                        ADC     %2              ;PROPOGATE CARRY
                                        ADD     #6,%2           ;ADD HIGHEST CONSTART
                                        ADD     %2,.AC          ;REPRIME R0 WITH HIGHEST DIGIT
                                        ADC     .MQ             ;PROPOGATE CARRY
                                        MOV     .AC,LONUM       ;SAVE R0
                                        MOV     .MQ,HINUM       ;SAVE R1
                                        MOV     (6)+,%4         ;RESTORE R4
                                        MOV     (6)+,%2         ;RESTORE R2
                                        MOV     (6)+,.MQ        ;RESTORE R1
                                        MOV     (6)+,.AC        ;RESTORE R0
                                        RTS     %7              ;RETURN


                        BAD:
                        64$:            TST     %3              ;WAIT FOR OLD PRINTING SEQUENCE
                                        BNE     64$             ;       TO BE COMPLETED
                                        MOV     #ERNG,%3        ;INITIALIZE PRINT REGISTER
                                        BIS     #100,@TPS
                        65$:            TST     %3              ;WAIT FOR OLD PRINTING SEQUENCE
                                        BNE     65$             ;       TO BE COMPLETED
                                        MOV     #CON,%3         ;INITIALIZE PRINT REGISTER
                                        BIS     #100,@TPS
                        66$:            TST     %3              ;WAIT FOR THE PRINTER TO FINISH
                                        BNE     66$
                                        HALT
                                        JSR     %7,CKSWR        ;CHECK FOR CONT-G
                                        RTS     %7
                        .TRAP:          MOV     @#177776,TAD    ;GET TRAP ADDRESS + 2
                                        CMP     #400,%6         ;CHECK FOR LIST OVERFLOW
                                        BMI     1$
                        64$:            TST     %3              ;WAIT FOR OLD PRINTING SEQUENCE
                                        BNE     64$             ;       TO BE COMPLETED
                                        MOV     #LSTERR,%3      ;INITIALIZE PRINT REGISTER
                                        BIS     #100,@TPS
                                        HALT                    ;HALT HERE MEANS PUSH DOWN LIST OVERFLOW
                                        JSR     %7,CKSWR        ;CHECK FOR CONT-G
                        1$:             CLC
                                        ROL     TAD             ;ROTATE TWICE LEFT
                                        ROL     TAD
                                        MOV     (6),TLOC        ;GET RETURN ADDRESS
                                        SUB     #2,TLOC         ;ADDRESS OF CAUSE
                        65$:            TST     %3              ;WAIT FOR OLD PRINTING SEQUENCE
                                        BNE     65$             ;       TO BE COMPLETED
                                        MOV     #TMES1,%3       ;INITIALIZE PRINT REGISTER
                                        BIS     #100,@TPS
                                        MOV     TAD,PRINT1      ;TYPE TAD IN OCTAL
                                        JSR     %7,PRINTS       ;AND SUPRESS LEADING ZERO'S
                        66$:            TST     %3              ;WAIT FOR OLD PRINTING SEQUENCE
                                        BNE     66$             ;       TO BE COMPLETED
                                        MOV     #TMES2,%3       ;INITIALIZE PRINT REGISTER
```

```
                                                BIS     #100,@TPS
                                                MOV     TLOC,PRINT1     ;TYPE TLOC IN OCTAL
                                                JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
                                        67$:    TST     %3              ;WAIT FOR OLD PRINTING SEQUENCE
                                                BNE     67$             ;       TO BE COMPLETED
                                                MOV     #TMES3,%3       ;INITIALIZE PRINT REGISTER
                                                BIS     #100,@TPS
                                                MOV     @TLOC,PRINT1    ;TYPE @TLOC IN OCTAL
                                                JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
                                        68$:    TST     %3              ;WAIT FOR OLD PRINTING SEQUENCE
                                                BNE     68$             ;       TO BE COMPLETED
                                                MOV     #CON,%3         ;INITIALIZE PRINT REGISTER
                                                BIS     #100,@TPS
                                        69$:    TST     %3              ;WAIT FOR THE PRINTER TO FINISH
                                                BNE     69$
                                                HALT                    ;HALT
                                                JSR     %7,CKSWR        ;CHECK FOR CONT-G
                                                RTI                     ;RETURN

                                        .TRT:   RTI
                                        TMES1:  .ASCII  "←←TRAP TO LOCATION &"
                                        TMES2:  .ASCII  " FROM &"
                                        TMES3:  .ASCII  " BY &"
                                        CON:    .ASCII  "←HIT CONTINUE TO PROCEDE←&"
                                        LSTERR: .ASCII  "←PUSH DOWN LIST OVERFLOW ←&"
                                        HEAD:   .ASCII  "← KE11 RANDOM EXERCISER ←&"
                                        MAIN:   .ASCII  "← MAINDEC-11-DZKEC-A ←&"
                                        EAEOK:  .ASCII  "← EAE OK ←&"
```

```
                    004720                          .EVEN
004720  000000      TAD:    .WORD   0               ;TRAP ADDRESS
004722  000000      TLOC:   .WORD   0               ;TRAP LOCATION
004724  000000      LONUM:  .WORD   0               ;RANDOM NUMBER #1
004726  000000      HINUM:  .WORD   0               ;RANDOM NUMBER #2
004730  000000      CINT:   .WORD   0               ;NUMBER OF TIMES IN SPECIFIC LOOP
004732  000000      CNT:    .WORD   0               ;PASS NUMBER
004734  000000      AREA:   .WORD   0               ;MQ (MULTIPLY)          DIVIDEND RIGHT
004736  000000                      0               ;AC (MULTIPLY)          DIVIDEND LEFT
004740  000000                      0               ;SR (MULTIPLY)          DIVISOR
004742  000000                      0               ;AC (DIVIDE BY LONUM)
004744  000000                      0               ;MQ (DIVIDE BY LONUM)
004746  000000                      0               ;SR (DIVIDE BY LONUM)
004750  000000                      0               ;AC (DIVIDE BY HINUM)
004752  000000                      0               ;MQ (DIVIDE BY HINUM)
004754  000000                      0               ;SR (DIVIDE BY HINUM)
004756  000000      ERCNT:  .WORD   0               ;ERROR COUNT
004760  000000      LAD:    .WORD   0               ;LOOP ADDRESS
004762  000000      .TYPE:  .WORD   0               ;CHARACTER TYPING LOCATION
004764  000000      ERRORS: .WORD   0               ;ERRORS ENCOUNTERED

004766  105713      TYPER:  TSTB    (3)             ;CHECK FOR NULL CHARACTER
004770  001403              BEQ     1$              ;IF NOT, TYPE THE CHARACTER
004772  122713  000046      CMPB    #'&,(3)         ;CHECK FOR TERMINATOR
004776  001005              BNE     .EMPTY
005000  042777  000100  173226  1$:   BIC    #100,@TPS
005006  005003              CLR     %3              ;CLEAR POINTER TO CHARACTER
005010  000002              RTI                     ;RETURN
005012  122713  000137      .EMPTY: CMPB    #'^,(3)         ;CRLF CHAR?
005016  001406              BEQ     .RET
005020  122713  000041              CMPB    #'!,(3)         ;CHECK FOR RETURN TERMINATOR
005024  001411              BEQ     .REST
005026  112377  173204              MOVB    (3)+,@TPB       ;TYPE CHARACTER
005032  000002              RTI                     ;RETURN
005034  005203      .RET:   INC     %3
005036  010367  000020              MOV     %3,.SAV         ;SET UP NEW POINTER
005042  012703  005056              MOV     #.RETR,%3
005046  000767              BR      .RET-6
005050  016703  000006      .REST:  MOV     .SAV,%3
005054  000744              BR      TYPER

005056  015     012     041 .RETR:  .BYTE   15,12,'!
                005062              .EVEN
005062  000000      .SAV:   0
```

```
                                ;;*****************************************************************
                                ;       OCTAL DUMP OF A WORD
                                ;;*****************************************************************


005064  000000                  PRINT1: 0
005066  000000  000000  000000  PRINT2: .WORD   0,0,0,0
005074  000000
005076     000     000          PRINT3: .BYTE   0,0

005100  112767  000001  177770  PRINTR: MOVB    #1,PRINT3       ;SET ZERO FILL SWITCH
005106  000402                  BR      PRNTCM
005110  005067  177762          PRINTS: CLR     PRINT3          ;SUPRESS LEADING ZERO'S
005114  112767  177772  177755  PRNTCM: MOVB    #-6,PRINT3+1    ;SET COUNT
005122  010546                  MOV     %5,-(6)         ;SAVE R5
005124  012705  005066          MOV     #PRINT2,%5      ;SET POINTER TO FIRST ASCII CHAR.
005130  105015                  CLRB    (5)             ;CLEAR FIRST BYTE
005132  000407                  BR      PRINTF          ;ROTATE FIRST BIT
005134  105015                  PRINTL: CLRB    (5)             ;CLEAR BYTE OF CHARACTER
005136  006167  177722          ROL     PRINT1          ;ROTATE BIT INTO C
005142  106115                  ROLB    (5)             ;PACK IT
005144  006167  177714          ROL     PRINT1          ;ROTATE BIT INTO C
005150  106115                  ROLB    (5)             ;PACK IT
005152  006167  177706          PRINTF: ROL     PRINT1          ;ROTATE BIT INTO C
005156  106115                  ROLB    (5)             ;PACK IT
005160  105715                  TSTB    (5)
005162  001402                  BEQ     1$
005164  105267  177706          INCB    PRINT3
005170  105767  177702          1$:     TSTB    PRINT3          ;CHECK FILL SWITCH
005174  001402                  BEQ     2$
005176  152725  000060          BISB    #'0,(5)+        ;MAKE INTO ASCII CHAR
005202  105267  177671          2$:     INCB    PRINT3+1
005206  001352                  BNE     PRINTL          ;REPEAT
005210  022705  005066          CMP     #PRINT2,%5
005214  001002                  BNE     3$
005216  112725  000060          MOVB    #'0,(5)+
005222  105015                  3$:     CLRB    (5)
005224  005703                  64$:    TST     %3              ;WAIT FOR OLD PRINTING SEQUENCE
005226  001376                  BNE     64$             ;       TO BE COMPLETED
005230  012703  005066          MOV     #PRINT2,%3      ;INITIALIZE PRINT REGISTER
005234  052777  000100  172772  BIS     #100,@TPS
005242  012605                  MOV     (6)+,%5         ;RESTORE R5
005244  000207                  RTS     %7
```

```
1036                                    ;;*****************************************************************
1037                                    ;       DIVIDE SUBROUTINE
1038                                    ;       R0,R1 CONTAIN AC,MQ RESPECTIVELY
1039                                    ;;*****************************************************************
1040
1041
1042
1043  005246  000000                    Z:      .WORD   0
1044  005250  000000                    COUNT:  .WORD   0
1045  005252  000000                    SR1:    .WORD   0
1046  005254  000000                    SINE:   .WORD   0
1047
1048
1049  005256  012767  177760  177764   DIVI:   MOV     #-16.,COUNT     ;SET UP STEP COUNTER
1050  005264  005067  177762                   CLR     SR1
1051  005270  005067  177760                   CLR     SINE
1052  005274  005700                            TST     .AC
1053  005276  100002                            BPL     STEP1
1054  005300  105167  177750                   COMB    SINE
1055  005304  005067  177736           STEP1:  CLR     Z               ;INITIALIZE Z INDICATORS
1056  005310  000241                            CLC
1057  005312  005700                            TST     .AC             ;CHECK REMAINDER SIGN
1058  005314  100002                            BPL     1$
1059  005316  105167  177725                   COMB    Z+1
1060  005322  005702                    1$:     TST     %2              ;CHECK DIVISOR SIGN
1061  005324  100002                            BPL     2$
1062  005326  105167  177715                   COMB    Z+1
1063  005332  105767  177711           2$:     TSTB    Z+1             ;-1 IF DIFFERENT, 0 IF SAME
1064  005336  100403                            BMI     3$
1065  005340  005267  177702                   INC     Z               ;SET Z=1 IF SAME
1066  005344  000261                            SEC                     ;SET C TO A 1
1067  005346  006101                    3$:     ROL     .MQ             ;ROTATE AC AND MQ ONCE LEFT  STEP2
1068  005350  006100                            ROL     .AC
1069  005352  105767  177670                   TSTB    Z               ;STEP3
1070  005356  001406                            BEQ     4$
1071  005360  160200                            SUB     %2,.AC          ;SUBTRACT IF Z = 1
1072  005362  106167  177667                   ROLB    SINE+1
1073  005366  105167  177663                   COMB    SINE+1
1074  005372  000403                            BR      5$
1075  005374  060200                    4$:     ADD     %2,.AC          ;ADD IF Z = 0
1076  005376  106167  177653                   ROLB    SINE+1
1077  005402  022767  177760  177640   5$:     CMP     #-16.,COUNT     ;CHECK FOR FIRST TIME THROUGH    STEP4
1078  005410  001014                            BNE     STEP8
1079  005412  005700                    STEP6:  TST     .AC
1080  005414  001412                            BEQ     STEP8
1081  005416  106067  177633                   RORB    SINE+1
1082  005422  103404                            BCS     1$              ;IF C AND DIVIDEND ARE
1083  005424  105767  177624                   TSTB    SINE
1084  005430  100431                            BMI     OFLW            ;       DIFFERENT, AN OVERFLOW
1085  005432  000403                            BR      STEP8           ;       HAS OCCURRED
1086  005434  105767  177614           1$:     TSTB    SINE
1087  005440  100025                            BPL     OFLW
1088
```

```
 1089  005442  005267  177602          STEP8:  INC     COUNT
 1090  005446  001402                          BEQ     1$
 1091  005450  000167  177630                  JMP     STEP1
 1092
 1093  005454  005700                  1$:     TST     .AC             ;BEGINNING OF CORRECTION ROUTINE
 1094  005456  001031                          BNE     STEP9           ;IF THE REMAINDER IS ZERO,
 1095  005460  000261                  STEP10: SEC                     ;       SHIFT THE QUOTIENT ONCE
 1096  005462  006101                          ROL     .MQ             ;       FILLING WITH "1"
 1097  005464  116705  177564          STP9:   MOVB    SINE,%5
 1098  005470  005702                          TST     %2
 1099  005472  100001                          BPL     1$
 1100  005474  005105                          COM     %5
 1101  005476  005705                  1$:     TST     %5
 1102  005500  100403                          BMI     2$
 1103  005502  005701                          TST     .MQ
 1104  005504  100403                          BMI     OFLW
 1105  005506  000474                          BR      DONE
 1106  005510  005701                  2$:     TST     .MQ
 1107  005512  003472                  3$:     BLE     DONE
 1108  005514  105767  177534          OFLW:   TSTB    SINE
 1109  005520  100004                          BPL     1$
 1110  005522  052767  000100  177522          BIS     #100,SR1
 1111  005530  000470                          BR      DONE1
 1112  005532  052767  000200  177512  1$:     BIS     #200,SR1
 1113  005540  000464                          BR      DONE1
 1114
 1115
 1116  005542  010267  177502          STEP9:  MOV     %2,COUNT
 1117  005546  100402                          BMI     1$
 1118  005550  005467  177474                  NEG     COUNT           ;IF THE DIVISOR WAS (+) NEGATE IT
 1119  005554  020067  177470          1$:     CMP     .AC,COUNT       ;IF THE SAME AS REMAINDER, GO
 1120  005560  001416                          BEQ     STEP11          ;       TO STEP 11
 1121
 1122  005562  105067  177461                  CLRB    Z+1
 1123  005566  005700                          TST     .AC             ;COMPARE THE SIGNS OF THE
 1124  005570  100002                          BPL     2$              ;       REMAINDER AND THE
 1125  005572  105167  177451                  COMB    Z+1             ;       DIVIDEND.  IF THEY
 1126  005576  105767  177452          2$:     TSTB    SINE            ;       ARE THE SAME, GO TO
 1127  005602  100002                          BPL     3$              ;       STEP 10
 1128  005604  105167  177437                  COMB    Z+1
 1129  005610  105767  177433          3$:     TSTB    Z+1
 1130  005614  001721                          BEQ     STEP10
```

```
1131  005616  005067  177424          STEP11: CLR     Z
1132  005622  005700                          TST     .AC             ;SET Z=1 IF REMAINDER AND
1133  005624  100002                          BPL     1$              ;       THE DIVISOR HAVE THE
1134  005626  105167  177415                  COMB    Z+1             ;       SAME SIGN
1135  005632  005702                  1$:     TST     %2
1136  005634  100002                          BPL     2$
1137  005636  105167  177405                  COMB    Z+1
1138  005642  105767  177401          2$:     TSTB    Z+1
1139  005646  001003                          BNE     3$
1140  005650  005267  177372                  INC     Z
1141  005654  005201                          INC     .MQ             ;ADD Z TO QUOTIENT
1142  005656  000241                  3$:     CLC
1143  005660  006101                          ROL     .MQ             ;SHIFT QUOTIENT LEFT ONCE
1144                                                                  ;       FILLING WITH "0"
1145  005662  105767  177360                  TSTB    Z
1146  005666  001402                          BEQ     4$
1147  005670  160200                          SUB     %2,.AC          ;SUBTRACT DIVISOR FROM REMAINDER
1148  005672  000674                          BR      STP9
1149  005674  060200                  4$:     ADD     %2,.AC          ;ADD DIVISOR TO REMAINDER
1150  005676  000672                          BR      STP9
1151
1152  005700  005701                  DONE:   TST     .MQ
1153  005702  100003                          BPL     DONE1
1154  005704  052767  000300  177340          BIS     #300,SR1
1155  005712  005700                  DONE1:  TST     .AC             ;TEST FOR AC=0
1156  005714  001010                          BNE     XX
1157  005716  005701                          TST     .MQ             ;TEST FOR MQ=0
1158  005720  001003                          BNE     1$
1159  005722  052767  000004  177322          BIS     #4,SR1          ;SET BIT 2 IF AC=MQ=0
1160  005730  052767  000020  177314  1$:     BIS     #20,SR1         ;SET BIT 4 IF AC=0
1161  005736  005701                  XX:     TST     .MQ             ;TEST FOR MQ=0
1162  005740  001003                          BNE     1$
1163  005742  052767  000010  177302          BIS     #10,SR1         ;SET BIT 3 IF MQ=0
1164  005750  005701                  1$:     TST     .MQ             ;GET SIGN OF MQ
1165  005752  100007                          BPL     ORV
1166  005754  022700  177777                  CMP     #-1,.AC         ;TEST FOR SINGLE PRECISION
1167  005760  001011                          BNE     NXT
1168  005762  052767  000042  177262          BIS     #42,SR1         ;SET BITS 1 AND 5 IF SINGLE
1169  005770  000405                          BR      NXT             ;       PRECISION NEGATIVE NUMBER
1170  005772  005700                  ORV:    TST     .AC
1171  005774  001003                          BNE     NXT
1172  005776  052767  000002  177246          BIS     #2,SR1          ;SET BIT 1 IF SINGLE PRECISION
1173                                                                  ;       POSITIVE NUMBER
1174  006004  022700  177777          NXT:    CMP     #-1,.AC
1175  006010  001003                          BNE     1$
1176  006012  052767  000040  177232          BIS     #40,SR1
1177  006020  000207                  1$:     RTS     %7
```

```
                                ;;*****************************************************************
                                ;       MULTIPLY SUBROUTINE
                                ;       R0,R1 CONTAINS AC,MQ RESPECTIVILY
                                ;;*****************************************************************



006022  000000          CONT:   .WORD   0
006024  000000          ONE:    .WORD   0
006026  000000          CHEK:   .WORD   0

006030  005067  177770  MUL1:   CLR     ONE             ;CLEAR Z STEP1
006034  005067  177212          CLR     SR1
006040  005067  177762          CLR     CHEK
006044  005000                  CLR     .AC             ;CLEAR AC
006046  005701                  TST     .MQ
006050  100002                  BPL     1$
006052  005167  177750          COM     CHEK
006056  005702          1$:     TST     %2
006060  100002                  BPL     2$
006062  005167  177740          COM     CHEK
006066  012767  177760  177726  2$:     MOV     #-16.,CONT      ;INITIALIZE COUNT
006074  022701  100000          CMP     #100000,.MQ
006100  001011                  BNE     CNXT
006102  005702                  TST     %2
006104  100023                  BPL     STEP2
006106  010200                  MOV     %2,.AC
006110  005400                  NEG     .AC
006112  000241                  CLC
006114  005001                  CLR     .MQ
006116  006000                  ROR     .AC
006120  006001                  ROR     .MQ
006122  000451                  BR      DONE2
006124  022702  100000  CNXT:   CMP     #100000,%2
006130  001011                  BNE     STEP2
006132  005701                  TST     .MQ
006134  100007                  BPL     STEP2
006136  010100                  MOV     .MQ,.AC
006140  005400                  NEG     .AC
006142  000241                  CLC
006144  005001                  CLR     .MQ
006146  006000                  ROR     .AC
006150  006001                  ROR     .MQ
006152  000435                  BR      DONE2
006154  105067  177645  STEP2:  CLRB    ONE+1
006160  132767  000001  177636  BITB    #1,ONE
```

```
1225  006166  001402                        BEQ     1$
1226  006170  105167  177631                COMB    ONE+1           ;COMP ONE+1 IF Z=1
1227  006174  032701  000001        1$:     BIT     #1,.MQ
1228  006200  001402                        BEQ     2$
1229  006202  105167  177617                COMB    ONE+1           ;COMP ONE+1 IF R1 IS ODD
1230  006206  105767  177613        2$:     TSTB    ONE+1           ;IF THE LOW ORDER BIT IS THE SAME
1231  006212  001406                        BEQ     STEP4           ;AS Z, GO TO STEP4
1232  006214  032701  000001        STEP3:  BIT     #1,.MQ
1233  006220  001002                        BNE     1$
1234  006222  060200                        ADD     %2,.AC          ;ADD MULTIPLICAND IF MQ IS EVEN
1235  006224  000401                        BR      STEP4
1236  006226  160200                1$:     SUB     %2,.AC          ;SUBTRACT MULTIPLICAND IF MQ IS ODD
1237  006230  006200                STEP4:  ASR     .AC
1238  006232  006001                        ROR     .MQ
1239  006234  106167  177564                ROLB    ONE
1240
1241  006240  005267  177556        STEP5:  INC     CONT
1242  006244  001343                        BNE     STEP2           ;GO TO STEP2 IF NOT 16TH PASS
1243  006246  005700                DONE2:  TST     .AC
1244  006250  001416                        BEQ     DO2
1245  006252  005767  177550                TST     CHEK
1246  006256  100406                        BMI     DON1
1247  006260  005700                        TST     .AC
1248  006262  100011                        BPL     DO2
1249  006264  005401                        NEG     .MQ
1250  006266  005500                        ADC     .AC
1251  006270  005400                        NEG     .AC
1252  006272  000405                        BR      DO2
1253  006274  005700                DON1:   TST     .AC
1254  006276  100403                        BMI     DO2
1255  006300  005401                        NEG     .MQ
1256  006302  005500                        ADC     .AC
1257  006304  005400                        NEG     .AC
1258  006306  005700                DO2:    TST     .AC
1259  006310  100200                        BPL     DONE1
1260  006312  052767  000300  176732        BIS     #300,SR1
1261  006320  000167  177366                JMP     DONE1
1262
1263
```

```
1264                                    ;;**********************************************************************
1265                                    ;       CHECK SWITCH REGISTER ROUTINE. CHECKS FOR ↑G TO ALLOW CHANGING
1266                                    ;       OF LOC.176.
1267                                    ;;**********************************************************************
1268
1269
1270  006324  000000                    JEMPST: .WORD   0
1271  006326  000000                    CONTT:  .WORD   0
1272  006330  000000                    RDSW:   .WORD   0
1273  006332  000000                    TIB:    .WORD   0
1274  006334  000000                    TOB:    .WORD   0
1275  006336  000000                    TOG:    .WORD   0
1276
1277
1278  006340  022767  000176  171660   CKSWR:  CMP     #SWREG,SWR      ;SOFTWARE SWITCH REGISTER PRESENT
1279  006346  001157                            BNE     OUT             ;NO, GET OUT
1280  006350  105777  171654                    TSTB    @TKS            ;YES,WAIT FOR
1281  006354  100154                            BPL     OUT             ;READY, GET CHARACTER
1282  006356  017767  171650  177746           MOV     @TKB,TIB        ;AND STRIP OFF
1283  006364  042767  177600  177740           BIC     #177600,TIB     ;THE GARBAGE
1284  006372  022767  000007  177732           CMP     #7,TIB          ;IS IT A <↑G>
1285  006400  001142                            BNE     OUT
1286  006402  005703                    64$:    TST     %3              ;WAIT FOR OLD PRINTING SEQUENCE
1287  006404  001376                            BNE     64$             ;       TO BE COMPLETED
1288  006406  012703  006762                    MOV     #$CNTG,%3       ;INITIALIZE PRINT REGISTER
1289  006412  052777  000100  171614           BIS     #100,@TPS
1290  006420                            CNTLU:
1291  006420  005703                    64$:    TST     %3              ;WAIT FOR OLD PRINTING SEQUENCE
1292  006422  001376                            BNE     64$             ;       TO BE COMPLETED
1293  006424  012703  006766                    MOV     #$MSWR,%3       ;INITIALIZE PRINT REGISTER
1294  006430  052777  000100  171576           BIS     #100,@TPS
1295  006436  017767  171564  176420           MOV     @SWR,PRINT1     ;TYPE @SWR IN OCTAL
1296  006444  004767  176430                    JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
1297  006450  005703                    65$:    TST     %3              ;WAIT FOR OLD PRINTING SEQUENCE
1298  006452  001376                            BNE     65$             ;       TO BE COMPLETED
1299  006454  012703  006775                    MOV     #$MNEW,%3       ;INITIALIZE PRINT REGISTER
1300  006460  052777  000100  171546           BIS     #100,@TPS
1301  006466  005037  006324                    CLR     @#TEMPST
1302  006472  005067  177626            $READ:  CLR     TEMPST
1303  006476  012767  000007  177622           MOV     #7,CONTT
1304  006504  004767  000200            1$:     JSR     %7,TTIN         ;GO  READ A CHARACTER
1305  006510  042767  177600  177614           BIC     #177600,TIB     ;STRIP OFF GARBAGE
1306  006516  122767  000025  177606           CMPB    #25,TIB         ;IS IT A ↑U?
1307  006524  001001                            BNE     2$              ;BRANCH IF NOT
1308  006526  000734                    3$:     BR CNTLU                ;START OVER
1309  006530  122767  000015  177574    2$:     CMPB    #15,TIB         ;IS IT A <CR>?
1310  006536  001017                            BNE     4$              ;BRANCH IF NOT
1311  006540  012767  000200  177562           MOV     #200,RDSW
1312  006546  005703                    64$:    TST     %3              ;WAIT FOR OLD PRINTING SEQUENCE
1313  006550  001376                            BNE     64$             ;       TO BE COMPLETED
1314  006552  012703  007011                    MOV     #$CRLF,%3       ;INITIALIZE PRINT REGISTER
1315  006556  052777  000100  171450           BIS     #100,@TPS
1316  006564  022767  000007  177534           CMP     #7,CONTT        ;WAS IT FIRST CHARACTER
1317  006572  001041                            BNE     7$              ;CHANGE SWR IF NOT FIRST ONE
1318  006574  000444                    8$:     BR      OUT             ;GET OUT
1319  006576  122767  000060  177526    4$:     CMPB    #60,TIB
```

```
1320  006604  003004                        BGT     5$
1321  006606  122767  000067  177516        CMPB    #67,TIB
1322  006614  002010                        BGE     6$
1323  006616                          5$:
1324  006616  005703                  65$:  TST     %3              ;WAIT FOR OLD PRINTING SEQUENCE
1325  006620  001376                        BNE     65$             ;       TO BE COMPLETED
1326  006622  012703  007005                MOV     #$QUEST,%3      ;INITIALIZE PRINT REGISTER
1327  006626  052777  000100  171400        BIS     #100,@TPS
1328  006634  000734                        BR      3$              ;START OVER IF NOT LEGAL CHARACTER
1329  006636  006367  177462          6$:   ASL     TEMPST
1330  006642  006367  177456                ASL     TEMPST
1331  006646  006367  177452                ASL     TEMPST
1332  006652  142767  000060  177452        BICB    #60,TIB         ;GET NITTY-GRITTY
1333  006660  156767  177446  177436        BISB    TIB,TEMPST
1334  006666  005367  177434                DEC     CONTT           ;ONLY WANT 6   DIGITS
1335  006672  001751                        BEQ     5$
1336  006674  000703                        BR      1$
1337  006676  016777  177422  171322  7$:   MOV     TEMPST,@SWR     ;CHANGE SWITCH REGISTER CONTENTS
1338  006704  000733                        BR      8$
1339  006706  000207                  OUT:  RTS     PC              ;RETURN TO PROGRAM
1340
1341                                  ;;**********************************************************************
1342                                  ;       TTY READ SUBROUTINE**********
1343                                  ;;**********************************************************************
1344
1345
1346
1347
1348  006710  005077  171314          TTIN:  CLR     @TKS
1349  006714  005077  171312                 CLR     @TKB
1350  006720  005067  177406                 CLR     TIB
1351  006724  005277  171300                 INC     @TKS
1352  006730  105777  171274          TTIN1: TSTB    @TKS
1353  006734  100375                        BPL     TTIN1
1354  006736  017767  171270  177366        MOV     @TKB,TIB
1355  006744  105777  171264          TTIN2: TSTB    @TPS
1356  006750  100375                        BPL     TTIN2
1357  006752  116777  177354  171256        MOVB    TIB,@TPB
1358  006760  000207                        RTS     %7
1359
1360  006762  057137  023107          $CNTG: .ASCII  "←↑G&"
1361  006766  051537  051127  020075  $MSWR: .ASCII  "←SWR= &"
1362  006774     046
1363  006775     040  047040  053505  $MNEW: .ASCII  "  NEW= &"
1364  007002  020075     046
1365  007005     040  020077     046  $QUEST: .ASCII  " ? &"
1366  007011     137  023040          $CRLF: .ASCII  "← &"
1367                                        .EVEN
1368
1369  007014  000000                  OFL:   0                       ;FIRST CHAR FLAG
1370          000001                         .END
```

```
AC      000206          349#    465     483     495*    504     517*    526     550*    554     778
AREA    004734          413     437     458     476     493     515     536     548     724     740     956#
ASH     000224          356#
BAD     004176          432     435     451     454     864#
BEGIN   001000          346     369#
CHEK    006026         1188#   1192*   1196*   1199*   1245
CINT    004730          374*    456*    546*    600     602*    612*    614*    618     620     641*    642     644*    954#
CKSWR   006340          407     574     603     608     615     624     650     665     676     685     876     886     917
                       1278#
CLCI    002356          612#    623     628
CNT     004732          373*    565*    566     693     955#
CNTLU   006420          395    1290#   1308
CNXT    006124         1202    1212#
CON     004536          655     871     912     928#
CONT    006022         1186#   1200*   1241*
CONTT   006326         1271#   1303*   1316    1334*
COUNT   005250         1044#   1049*   1077    1089*   1116*   1118*   1119
DIV     000204          348#    501*    523*    551*
DIVI    005256          423     443     544    1049#
DIVP    003202          702     724#
DIVZ    003314          739     743#
DONE    005700         1105    1107    1152#
DONE1   005712         1111    1113    1153    1155#   1259    1261
DONE2   006246         1211    1222    1243#
DON1    006274         1246    1253#
DOWN    002572          650#
DO2     006306         1244    1248    1252    1254    1258#
EAEOK   004704          571     585     947#
EQZ     002276          593     595     597#
ERCNT   004756          611*    622     627*    634     640*    680     796*    965#
ERNG    003776          821#    867
ERRORS  004764          370*    684*    968#
ER1     003634          697     800#
ER2     003650          709     719     802#
ER2A    003654          736     803#
ER3     003660          754     804#
ER3A    003672          776     806#
ER3B    003702          772     808#
ER4     003714          760     782     810#
ER5     003724          766     788     812#
ER6     003734          632     814#
ER7     003757          638     818#
FLOW    002236          502     524     552     590#
GO      001164          407#    573     587
HEAD    004623          377     938#
HINUM   004726          405*    416     433     442     447     464     481     523     538     540     711     715     749
                        835     845     856*    953#
HLT   = 104400          365#    467     470     473     485     488     491     506     509     512     528     531     534
                        556     559     562
HLTAD   003630          683*    794     798#
ITER    001404          448     453     456#
KIT     002364          610     614#
LAD     004760          606*    663     966#
LONUM   004724          404*    415     422     428     452     463     482     501     537     539     542     543     705
                        721     745     834     843     855*    952#
LSH     000222          355#
```

```
LSTERR  004570            883     933#
MAIN    004655            381     943#
MQ      000210            350#    463*    468     481*    486     494*    507     516*    529     549*    557     784
MUL     000212            351#    464*    482*
MUL1    006030            417    1190#
NOHEAD  003614            679     682     794#
NOR     000220            354#
NXT     006004           1167    1169    1171    1174#
NXT1    002336            601     608#
OFL     007014           1369#
OFLW    005514           1084    1087    1104    1108#
OK1     001320            429     434     437#
ONE     006024           1187#   1190*   1223*   1224    1226*   1229*   1230    1239*
ORV     005772           1165    1170#
OUT     006706           1279    1281    1285    1318    1339#
OVR     002634            605     617     621     661#
PC    =%000007            395*   1339*
PRINTF  005152           1010    1016#
PRINTL  005134           1011#   1025
PRINTR  005100            706     712     716     722     726     733     741     746     750     757     763     779     785
                          903     909    1003#   1296
PRINTS  005110            635     643     694     769     792     897    1005#
PRINT1  005064            634*    642*    693*    705*    711*    715*    721*    725*    732*    740*    745*    749*    756*
                          762*    768*    778*    784*    791*    896*    902*    908*    998#   1012*   1014*   1016*   1295*
PRINT2  005066            999#   1008    1026    1032
PRINT3  005076           1001#   1003*   1005*   1006*   1020*   1021    1024*
PRNTCM  005114           1004    1006#
PSOFT   003352            713     723     742     747     751#
RANDOM  004052            414     541     830#
RDSW    006330           1272#   1311*
RT      002630            652     660#
SC      000214            352#
SCOPE = 104000            364#    457     475     492     514     535     547     563
SECI    003340            744     748#
SECO    003136            704     714#
SER     003632            790*    791     799#
SHIFT   004100            838#    842
SINE    005254           1046#   1051*   1054*   1072*   1073*   1076*   1081*   1083    1086    1097    1108    1126
SP    =%000006            383*    384*    389     391     392
SR      000216            353#    471     489     510     532     560     590     790
SR1     005252            420     426     446     461*    471     479*    489     499*    510     521*    532     560     768
                         1045#   1050*   1110*   1112*   1154*   1159*   1160*   1163*   1168*   1172*   1176*   1191*   1260*
STEP1   005304           1053    1055#   1091
STEP10  005460           1095#   1130
STEP11  005616           1120    1131#
STEP2   006154           1204    1213    1215    1223#   1242
STEP3   006214           1232#
STEP4   006230           1231    1235    1237#
STEP5   006240           1241#
STEP6   005412           1079#
STEP8   005442           1078    1080    1085    1089#
STEP9   005542           1094    1116#
STP9    005464           1097#   1148    1150
SUSWR   001064            383#
SWR     000226            357#    386     390*    393     408     575     604     609     616     625     651     666     677
                          686    1278    1295    1337*
```

```
SWREG   000176          344#    390     393     1278
TAD     004720          878*    888*    889*    896     950#
TEMP    002300          590*    591*    592     594     599#
TEMPST  006324         1270#   1301*   1302*   1329*   1330*   1331*   1333*   1337
TIB     006332         1273#   1282*   1283*   1284    1305*   1306    1309    1319    1321    1332*   1333    1350*   1354*
                       1357
TKB     000232          359#   1282    1349*   1354
TKS     000230          358#   1280    1348*   1351*   1352
TKV   = 000060          362#
TLOC    004722          890*    891*    902     908     951#
TMES1   004476          894     921#
TMES2   004522          900     925#
TMES3   004531          906     927#
TOB     006334         1274#
TOG     006336         1275#
TPB     000236          361#    981*   1357*
TPS     000234          360#    378*    382*    572*    582*    586*    633*    639*    649*    656*    673*    692*    698*
                        710*    720*    731*    737*    755*    761*    767*    773*    777*    783*    789*    868*    872*
                        884*    895*    901*    907*    913*    974*   1033*   1289*   1294*   1300*   1315*   1327*   1355
TPV   = 000064          363#
TST1    002422          619     622#
TTIN    006710         1304    1348#
TTIN1   006730         1352#   1353
TTIN2   006744         1355#   1356
TYPER   004766          341     970#    988
XX      005736         1156    1161#
Z       005246         1043#   1055*   1059*   1062*   1063    1065*   1069    1122*   1125*   1128*   1129    1131*   1134*
                       1137*   1138    1140*   1145
$CNTG   006762         1288    1360#
$CRLF   007011         1314    1366#
$MNEW   006775         1299    1363#
$MSWR   006766         1293    1361#
$QUEST  007005         1326    1365#
$READ   006472         1302#
.     = 007016          331#    332#    335#    340#    343#    345#    368#    949#    991#
.AC   =%000000          366#    419     424     430     439*    444     449     460*    465     478*    483     497*    504
                        519*    526     540*    554     756     830     834*    838*    847*    853*    855     860*   1052
                       1057    1068*   1071*   1075*   1079    1093    1119    1123    1132    1147*   1149*   1155    1166
                       1170    1174    1193*   1205*   1206*   1209*   1216*   1217*   1220*   1234*   1236*   1237*   1243
                       1247    1250*   1251*   1253    1256*   1257*   1258
.EMPTY  005012          973     977#
.EMT    002302          336     600#    613     660
.ET     002712          668     676#
.MQ   =%000001          367#    415*    418     425     433     438*    445     452     459*    468     477*    486     498*
                        507     520*    529     539*    557     762     831     835*    839*    844*    845*    848*    850*
                        854*    856     859*   1067*   1096*   1103    1106   1141*   1143*   1152    1157    1161    1164
                       1194    1201    1208*   1210*   1214    1216    1219*   1221*   1227    1232    1238*   1249*   1255*
.REST   005050          980     987#
.RET    005034          978     983#    986
.RETR   005056          985     990#
.SAV    005062          984*    987     992#
.SR1    001644          503     508     510#
.SR2    001744          525     530     532#
.SR3    002100          553     558     560#
.TRAP   004246          878#
.TRP    002646          338     665#
```

.TRT 004474 333 920#
.TYPE 004762 578* 581 645* 648 669* 672 688* 691 727* 730 967#

```
COMMEN      1#
DUMP      368#    705     711     714     721     725     732     740     745     748     756     762     778     784     902
           908    1295
ENDCOM      1#
ESCAPE      1#
GETPRI      1#
GETSWR      1#
HOLD      368#    657     674     873     914
MULT        1#
NEWTST      1#
POP         1#
PUSH        1#
REPORT      1#
SDUMP     368#    634     642     693     768     791     896
SETPRI      1#
SETUP       1#
SKIP        1#
SLASH       1#
STARS       1#    299#    300     314     327     397     399     993     995    1036    1039    1178    1181    1264    1267
          1341    1343
SWRSU       1#
TYPBIN      1#
TYPDEC      1#
TYPE      368#    375     379     568     579     583     629     636     646     653     670     689     695     707     717
           728     734     751     758     764     770     774     780     786     864     869     881     892     898     904
           910    1030    1286    1290    1297    1312    1323
TYPEM     368#    578     645     669     688     727
TYPNAM      1#
TYPNUM      1#
TYPOCS      1#
TYPOCT      1#
TYPTXT      1#
$$ESCA      1#
$$NEWT      1#
$$SKIP      1#
.EQUAT      1#
.HEADE      1#    299#    302
.KT11       1#
.SETUP      1#
.SWRHI      1#
.$ACT1      1#
.$APTB      1#
.$APTH      1#
.$APTY      1#
.$ASTA      1#
.$CATC      1#
.$CMTA      1#
.$DB2D      1#
.$DB2O      1#
.$DIV       1#
.$EOP       1#
.$ERRO      1#
.$ERRT      1#
.$MULT      1#
.$POWE      1#
.$RAND      1#
```

```
.$RDDE      1#
.$RDOC      1#
.$READ      1#
.$R2AZ      1#
.$SAVE      1#
.$SB2D      1#
.$SB2O      1#
.$SCOP      1#
.$SIZE      1#
.$SUPR      1#
.$TRAP      1#
.$TYPB      1#
.$TYPD      1#
.$TYPE      1#
.$TYPO      1#
.$4DCA      1#
.1170       1#
```

```
ADC      844   846   848   849   851   854  1250  1256
ADD      440   518   565   596   843   845   847   850   852   853  1075  1149  1234
ASL      838  1329  1330  1331
ASR     1237
BCS     1082
BEQ      387   429   431   434   448   450   453   466   469   472   484   497   490   505   508
         511   527   530   533   555   558   561   567   593   595   610   623   626   668   678
         681   704   739   971   978   980  1019  1022  1070  1080  1090  1120  1130  1146  1225
        1228  1231  1244  1335
BGE     1322
BGT     1320
BIC      974  1283  1305
BICB     591  1332
BIS      378   382   410   572   582   586   633   639   649   656   673   692   698   710   720
         731   737   755   761   767   773   777   783   789   868   872   884   895   901   907
         913  1033  1110  1112  1154  1159  1160  1163  1168  1172  1176  1260  1289  1294  1300
        1315  1327
BISB    1023  1333
BIT      408   566   575   604   609   616   625   666   677   686   701   703   738   743  1227
        1232
BITB    1224
BLE     1107
BMI      619   880  1064  1084  1102  1104  1117  1246  1254
BNE      376   380   394   409   570   577   580   584   601   605   617   631   637   647   654
         658   671   675   687   690   696   702   708   718   729   735   744   753   759   765
         771   775   781   787   795   842   866   870   874   882   893   899   905   911   915
         973  1025  1027  1031  1078  1094  1139  1156  1158  1162  1167  1171  1175  1202  1213
        1233  1242  1279  1285  1287  1292  1298  1307  1310  1313  1317  1325
BPL      621   652  1053  1058  1061  1087  1099  1109  1124  1127  1133  1136  1153  1165  1195
        1198  1204  1215  1248  1259  1281  1353  1356
BR       388   411   503   525   553   613   628   713   723   742   747   986   988  1004  1010
        1074  1085  1105  1111  1113  1148  1150  1169  1211  1222  1235  1252  1308  1318  1328
        1336  1338
CLC      887  1056  1142  1207  1218
CLR      370   371   372   374   412   456   546   611   612   627   640   644   837   975  1005
        1050  1051  1055  1131  1190  1191  1192  1193  1208  1219  1301  1302  1348  1349  1350
CLRB    1009  1011  1029  1122  1223
CMP      386   389   393   433   452   465   468   483   486   504   507   526   529   554   557
         620   794   879  1026  1077  1119  1166  1174  1201  1212  1278  1284  1316
CMPB     471   489   510   532   560   592   594   972   977   979  1306  1309  1319  1321
COM     1100  1196  1199
COMB    1054  1059  1062  1073  1125  1128  1134  1137  1226  1229
DEC      641  1334
EMT      364
HALT     332   659   875   885   916
INC      602   614   684   796   841   983  1065  1089  1140  1141  1241  1351
INCB    1020  1024
JMP      346   573   587   660   663   679   682  1091  1261
JSR      395   407   414   417   423   432   435   443   451   454   502   524   541   544   552
         574   603   608   615   624   635   643   650   665   676   685   694   706   712   716
         722   726   733   741   746   750   757   763   769   779   785   792   876   886   897
         903   909   917  1296  1304
MOV      369   373   377   381   383   384   385   390   391   392   404   405   413   415   416
         418   419   420   422   424   425   426   437   438   439   442   444   445   446   458
         459   460   461   463   464   476   477   478   479   481   482   493   494   495   497
         498   499   501   515   516   517   519   520   521   523   536   537   538   539   540
```

```
        542   543   548   549   550   551   571   581   585   606   632   634   638   642   648
        655   662   672   683   691   693   697   699   705   709   711   715   719   721   724
        725   730   732   736   740   745   749   754   756   760   762   766   768   772   776
        778   782   784   788   791   830   831   832   833   834   835   836   855   856   857
        858   859   860   867   871   878   883   890   894   896   900   902   906   908   912
        984   985   987  1007  1008  1032  1034  1049  1116  1200  1205  1216  1282  1288  1293
       1295  1299  1303  1311  1314  1326  1337  1354
MOVB    578   590   645   669   688   727   790   981  1003  1006  1028  1097  1357
NEG    1118  1206  1217  1249  1251  1255  1257
ROL     839   840   888   889  1012  1014  1016  1067  1068  1096  1143
ROLB   1013  1015  1017  1072  1076  1239
ROR    1209  1210  1220  1221  1238
RORB   1081
RTI     607   797   918   920   976   982
RTS     597   861   877  1035  1177  1339  1358
SEC    1066  1095
SUB     700   891  1071  1147  1236
TRAP    365
TST     375   379   428   430   447   449   496   569   579   583   600   618   622   630   636
        646   651   653   657   661   670   674   680   689   695   707   717   728   734   752
        758   764   770   774   780   786   793   865   869   873   881   892   898   904   910
        914  1030  1052  1057  1060  1079  1093  1098  1101  1103  1106  1123  1132  1135  1152
       1155  1157  1161  1164  1170  1194  1197  1203  1214  1243  1245  1247  1253  1258  1286
       1291  1297  1312  1324
TSTB    970  1018  1021  1063  1069  1083  1086  1108  1126  1129  1138  1145  1230  1280  1352
       1355
.ABS    330
.ASCII  800   802   803   804   806   808   810   812   814   818   821   921   925   927   928
        933   938   943   947  1360  1361  1363  1365  1366
.BYTE   990  1001
.ENABL    1   299
.END   1370
.ENDC   302   307   315   329   398   400   994   996  1037  1040  1179  1182  1265  1268  1342
       1344
.EVEN   829   949   991  1367
.IF     300   303   314   327   397   399   993   995  1036  1039  1178  1181  1264  1267  1341
       1343
.IFF    300   315   327   398   400   994   996  1037  1040  1179  1182  1265  1268  1342  1344
.IIF    302   307
.LIST     1   299   300   301   302   327   328   329   332   368
.MACR   368
.MACRO    1   299
.NLIST    1   299   300   301   302   327   328   329   332   368
.REM      1
.REPT   300   327   332
.TITLE  302
.WORD   950   951   952   953   954   955   956   965   966   967   968   999  1043  1044  1045
       1046  1186  1187  1188  1270  1271  1272  1273  1274  1275
```

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

#,DZKECA.SEQ/SOL/CRF/PAGNUM/NL:TOC=SYSMAC.CO,DZKECA.P11

RUN-TIME: 24 23 2 SECONDS
RUN-TIME RATIO: 136/55=2.4
CORE USED: 33K (65 PAGES)

M03

Spooler runtime 5 Seconds, 25 KCS, 133 disk reads, 3 disk writes, 37 pages